Journal of Computational and Applied Mathematics 12&13 (1985) 3–18
North-Holland

Incomplete block matrix factorization preconditioning methods. The ultimate answer?

O. AXELSSON
Department of Mathematics, Catholic University, Nijmegen, The Netherlands *

Received 23 September 1984

Abstract: A considerable interest has been devoted to block matrix incomplete factorization preconditioning methods in recent time, see [12,2,3,4,6,7,8,10 and 21].

The purpose of this paper is to survey some of these results and to present new results showing how much more efficient and robust such methods can be as compared to other, admittingly also efficient, methods such as pointwise incomplete factorization preconditioning methods and multilevel methods with hierarchical basis functions, both on scalar as well as on parallel or vectorizable computers.

1. Introduction

We shall consider the numerical solution of very large but sparse linear algebraic systems

$$Ax = b, \quad x, b \in \mathbb{R}^N. \tag{1.1}$$

Although the methods we shall present do not always require this, for ease of presentation we shall assume that A is a nonsingular M-matrix (i.e., $a_{ij} \leqslant 0$, $i \neq j$ and the entries of the inverse, A^{-1} are nonnegative).

Such matrices may, for instance, arise when we apply a difference method or the lowest order finite element method for a diffusion equation. They also arise at each correction step of a defect-correction method for convection-diffusion problems, where the correction operator is derived from an artificial diffusion or upwind difference (or finite element) operator.

The order N may be very large. For example, for a scalar equation in three space dimensions (3D) on a $64 \times 64 \times 64$ mesh we get $N \approx 250\,000$.

Direct solution methods for such problems suffer from fill-in to such an extent that on fine meshes they cannot be solved to a reasonable cost even on presently available supercomputers.

Iterative methods on the other hand do not suffer from fill-in and with effective (modified) preconditioned and accelerated iterative methods one may derive algorithms of almost optimal order of computational complexity.

* The research reported in this paper was supported by the North Atlantic Treaty Organization, Brussels, through Grant No. 648/83.

Traditional preconditioned methods are based on incomplete (pointwise) factorizations of the given sparse matrix and result in quite efficient algorithms on a scalar computer, see [23,25,19,16 and 5]. These methods are however not vectorizable in their original form.

Newly developed versions of approximate factorization methods based on approximations of the inverses of diagonal block matrices, are however vectorizable and parallelizable to a large extent and are in fact also more efficient than their pointwise counterparts even on scalar computers as we shall see. Such a method is presented here and compared to other methods. It is found that even on quite fine meshes the method performs as well as certain multilevel methods of almost optimal asymptotic order of computational complexity (on a scalar computer). Furthermore, because of a superlinear rate of convergence, they may eventually outperform them even on the finest meshes when a sufficiently small iteration error is requested. For a recent discussion about vector computations for sparse linear systems, see [18].

In Section 2 we survey recent results about the existence of such factorizations and present an almost fully parallel version of them.

In Section 3 we present two alternatives of the methods for matrices of the type which arises for difference equations in three space dimensions.

In Section 4 we present a modification of the methods, which is based on a generalized rowsum criterion, to decrease the associated spectral condition number by an order of magnitude. We also discuss a new, relaxed form of modification, recently proposed in [9].

In Section 5 we show results about the distribution of eigenvalues of the preconditioned matrix and find an interesting clustering behavior, in particular for the relaxed methods. This is shown to lead to a superlinear rate of convergence when the preconditioning is coupled with a conjugate gradient acceleration method. Some numerical tests that support the theoretical results are also reported.

Finally, in Section 6 we comment on the robustness of the methods for problems such as with unisotropy and for convection dominated convection-diffusion problems.

2. Fully parallel or vectorizable variants of block matrix approximate factorization preconditionings

Let $C = LU$ be a sparse approximation, for instance, an incomplete factorization of A. Because of the advantageous effect C may have on the condition number of $C^{-1}A$ as compared to that of A, C is often called a preconditioning matrix.

A basic iterative method for the solution of (1.1) has the form

$$C\delta^{l+1} = -r^l, \quad x^{l+1} = x^l + \delta^{l+1}, \qquad l = 0, 1, 2, \ldots, \tag{2.1}$$

of a *defect-correction method* where $r^l = Ax^l - b$ is the defect or residual and δ^{l+1} is the correction of stage l. x^0 is arbitrary but a good choice is $x^0 = C^{-1}b$.

Consider the splitting $A = C - R$ of A, where R is the defect matrix. R is sparse, frequently even much sparser than A. Then (2.1) takes the form

$$Cx^{l+1} = Rx^l + b, \quad l = 0, 1, 2, \ldots.$$

which converges if and only if $\rho(C^{-1}R) < 1$, where $\rho(\cdot)$ is the spectral radius. The theory of (quasi) regular splittings may be applied (see [24] and [11]). The rate of convergence as measured in the number of iterations to reach a relative error, $\|x - x^k\|/\|x - x^0\| \leqslant \varepsilon$ is $k \cong \ln(1/\varepsilon)/\ln(1/\rho_0)$, where $\rho_0 = \|C^{-1}R\|$.

For second order elliptic problems and if $C = D_A$, the (block) diagonal part of A, one gets $\rho_0 = 1/(1 + \zeta h^2)$, for some positive ζ, independent of h. Hence $k = O(h^{-2})$ which is unacceptable.

The efficiency of the basic iterative method may be improved in two ways:

(i) by a more efficient choice of the preconditioning matrix C,

(ii) by use of some accelerated form of iterative method such as some preconditioned and generalized conjugate gradient method.

This paper deals with the first topic. As far as the second is concerned we shall here just make the following comment.

For a symmetric positive definite (SPD) problem, one may prove that $\|x - x^k\|_{A^{1/2}} \leqslant \varepsilon \|x - x^0\|_{A^{1/2}}$ if

$$k = \tfrac{1}{2}\kappa^{1/2} \ln(2/\varepsilon) \tag{2.2}$$

where $\kappa = \max_i \lambda_i / \min_i \lambda_i$, the spectral condition number and $\|x\|_{A^{1/2}} = \{x^T A x\}^{1/2}$, the so-called energy norm. Here $\{\lambda_i\}$ is the set of eigenvalues of $C^{-1}A$. The number of iterations in (2.2) is an upper bound, frequently it is overly pessimistic. For an SPD problem with $C = D_A$ as before we get now $k = O(h^{-1})$, an order of improvement over the basic method. This may be further improved upon by use of so-called modified incomplete factorization methods (see [16] and [5]). One may prove that then (again for second order elliptic problems), $k = O(h^{-1/2})$. This results in a total cost of $O(N^{1.25})$, $d = 2$ and $O(N^{1/17})$, $d = 3$ where d is the space dimension. Storage is of optimal order, $O(N)$, typically we need about twice as much storage as needed for A alone. For time-dependent problems solved by implicit timestepping this may be further improved upon. Hence such methods have an almost optimal order of computational complexity.

We shall consider two versions of approximate factorizations (of A of the generalized SSOR type (cf. [1,2,7 and 8]). Let $A = D - L - U$ where L and U, respectively, are the block lower and upper triangular parts of A.

Version 1 (Diagonal block version):

$$C = (\tilde{D} - L)\tilde{D}^{-1}(\tilde{D} - U) = (\tilde{D} - L)(I - \tilde{D}^{-1}U).$$

Version 2 (Diagonal block inverses version or the division (inverse) free factorization):

$$C = (I - L\tilde{D})\tilde{D}^{-1}(I - \tilde{D}U) = (\tilde{D}^{-1} - L)(I - \tilde{D}U).$$

In both versions we want to choose $\tilde{D}$ so that C is in some sense a good approximation of A. In the classical SSOR method (see [26] and [5]) we let $\tilde{D} = D/\omega$ (applicable to Version 1) and try to choose ω, $0 < \omega < 2$ so that the spectral condition number $\kappa(C^{-1}A)$ is minimized (in the case A is symmetric and positive definite). Before we describe a more efficient choice of $\tilde{D}$ we make the following observations.

We note that in Version 1 we have to solve linear systems with the diagonal blocks of $\tilde{D}$ in the forward and backward substitution processes. But solving linear systems by standard direct methods is a sequential process, hence not well suited for parallel computers.

In Version 2, however, the diagonal blocks appear as inverses. Hence within the blocks only matrix-vector multiplications occur and this version is consequently more suitable for vectorizable and parallel computers.

However, there will still be a recursion among the blocks because we have to solve the linear block systems with matrices $\tilde{D}^{-1} - L$ and $I - \tilde{D}U$, which occur at every iteration step. This may

be done by the usual forward and backward substitution methods for block matrices. However, again since this is a sequential process, we describe now an algorithm where also this part may be done in parallel.

At every iteration step with the matrix C of Version 2 as preconditioner, we have to solve a linear system of the form $(I - L\tilde{D})\tilde{D}^{-1}(I - \tilde{D}U)x = b$, i.e., we have to calculate

$$x = (I - \tilde{D}U)^{-1}\tilde{D}(I - L\tilde{D})^{-1}b. \tag{2.3}$$

Now, if $\tilde{D}$ has n blocks, then a Neumann series expansion gives $(I - L\tilde{D})^{-1} = I + L\tilde{D} + (L\tilde{D})^2 + \cdots + (L\tilde{D})^{n-1}$ because $L\tilde{D}$ is a nilpotent (block) matrix of index n. Further, we notice that this finite sum may be written as a finite product

$$(I - L\tilde{D})^{-1} = (I + L\tilde{D})\big(I + (L\tilde{D})^2\big)\big(I + (L\tilde{D})^4\big)\cdots\big(I + (L\tilde{D})^{2^s}\big) \tag{2.4}$$

in only $s + 1$ factors, where $s = \lfloor \log_2 n \rfloor - 1$ and $\lfloor a \rfloor \geqslant a$ indicates the smallest integer not smaller than a. Similarly, we get

$$(I - \tilde{D}U)^{-1} = (I + \tilde{D}U)\big(I + (\tilde{D}U)^2\big)\cdots\big(I + (\tilde{D}U)^{2^s}\big). \tag{2.5}$$

Hence, by use of (2.4) and (2.5), the power (factor) expansion method, only matrix-vector multiplications occur in the calculation of x in (2.3) and this leads hence to *a fully vectorizable variant of preconditioning.*

Note that we may utilize parallelism in two ways. For each factor in (2.4) and (2.5) each block may be multiplied with the corresponding block in the partioned vector in parallel and within each block the multiplication of the matrix rows with this vector may be performed in parallel. (Frequently it is more efficient to work on subdiagonals instead of rows or columns). Hence the depth of this algorithm is $O(\log_2 n)$ (the number of factors in (2.4) and (2.5)).

The disadvantage with this method is that, although the matrix $L\tilde{D}(\tilde{D}U)$ may be sparse, the matrices $(L\tilde{D})^r$ (and $(\tilde{D}U)^r$) get increasingly full when r increases. To offset this it is advisable to approximate the factors by matrices with small bandwidths. That this is a reasonable thing to do is further supported by the fact that in many problems the entries of $(L\tilde{D})^r$ (and of $(\tilde{D}U)^r$) gets increasingly smaller as r increases. This will however not further be discussed in the present paper.

It is interesting to note that the Neumann series expansion, implemented by say Horners scheme, has recursiveness of length $n - 1$, whereas by use of (2.4) and (2.5), we get algorithms of depth[1] $\lfloor \log_2 n \rfloor$ (assuming that we approximate the factors as indicated above).

In the following we shall consider the case where A, an M-matrix, is given on tridiagonal block matrix form (has block property A, see [26]) such as the one one gets by use of a proper ordering for discretized partial differential equation on a plane domain ($d = 2$) or on a domain in three space dimensions ($d = 3$). We have then

$$A = A^{(d)} = \begin{bmatrix} A_{1,1}^{(d-1)} & A_{1,2} & & 0 \\ A_{2,1} & A_{2,2}^{(d-1)} & A_{2,3} & \\ & \ddots & \ddots & \\ 0 & & A_{n,n-1} & A_{n,n}^{(d-1)} \end{bmatrix}. \tag{2.6}$$

[1] The depth of an algorithm is the smallest recursive length of any version, performing the same computational task.

Here $A_{i,j}$, $i \neq j$ are matrices with a small bandwidth with nonzero entries located about their main diagonal. The diagonal blocks $A_{i,j}^{(d-1)}$, are 'line-blocks' in case $d = 2$, i.e., bandmatrices with a small halfbandwidth q (typically $q = 1$) located symmetrically about their main diagonal. In case $d = 3$, each diagonal block has the same structure as a two-dimensional matrix $A^{(2)}$.

In the following we shall delete the superindices, d and $d-1$, except when needed to clarify the algorithms to be presented. Note that in the case A has the form (2.6) the terms $(\tilde{D}U)^r$ in the factors of (2.5) (and correspondingly for the lower triangular terms $(L\tilde{D})^r$ in (2.4)) consists only of super- (and sub-) diagonal blocks (see [8]). In Version 2 as applied on A defined by (2.6) we have

$$C = \begin{bmatrix} \tilde{D}^{-1} & & & 0 \\ A_{2,1} & \tilde{D}_2^{-1} & & \\ & \ddots & \ddots & \\ 0 & & A_{n,n-1} & \tilde{D}_n^{-1} \end{bmatrix} \begin{bmatrix} I & (\tilde{D}_1 A_{1,2}) & & 0 \\ & I & (\tilde{D}_2 A_{2,3}) & \\ & & \ddots & \\ 0 & & & I \end{bmatrix}.$$

Note that the block-diagonal matrices in the first factor appear as inverses. This differs from the methods considered in [12] and in [2]. Our present modification leads to highly parallel algorithms, where we have avoided solutions of all linear systems with block-diagonal matrices. We get

$$C = \begin{bmatrix} \tilde{D}_1^{-1} & A_{1,2} & & 0 \\ A_{2,1} & (\tilde{D}_2^{-1} + A_{2,1}\tilde{D}_1 A_{1,2}) & A_{2,3} & \\ & \ddots & \ddots & \\ 0 & & A_{n,n-1} & (\tilde{D}_n^{-1} + A_{n,n-1}\tilde{D}_{n-1}A_{n-1,n}) \end{bmatrix}.$$

Consider the matrix recursion

$$\tilde{D}_0 = 0, \qquad \tilde{D}_i = (A_{i,i} - A_{i,i-1}\tilde{D}_{i-1}A_{i-1,i})^{-1}, \quad i = 1, 2, \ldots, n, \tag{2.7}$$

(2.7) will give a complete factorization of A, i.e., in this case $C = A$ and, in general, the matrices $\tilde{D}_i$ are full matrices. Hence we would need a storage for n full block matrices for the factorization and about $n(s+1)$ full block matrices for the fully vectorizable version, based on (2.4) and (2.5). This is clearly unacceptable for large sized problems. (It is interesting to note that if we store the matrices $\tilde{D}_i$ as well as $A_{i,i-1}$ and $A_{i,i+1}$, $i = 1, 2, \ldots, n$, the demand of storage is typically about half of that needed for a bandmatrix factorization, in case the matrices $A_{i,i-1}$ and $A_{i,i+1}$ are sparse. The complete factorization with the diagonal block matrices calculated by (2.7) is recommended as a direct method for small sized problems in particular for parallel or vectorizable computers.)

We shall now consider *sparse matrix recursions* where the correspondingly calculated factorized matrix shall be used as a preconditioner for some iterative method. Using M-matrix theory it is proved in [8] that there exists recursions which satisfy

$$0 \leqslant \tilde{D}_i \leqslant (A_{i,i} - A_{i,i-1}\tilde{D}_{i-1}A_{i-1,i})^{-1}, \quad i = 1, 2, \ldots, n \tag{2.8}$$

This is the recursion relation we use for Version 2. For the corresponding recursion which occurs in Version 1, existence results are found in [2] and [4].

We consider at first the following particular class of methods for the construction of a sparse sequence of matrices $\{\tilde{D}_i\}$ satisfying (2.8). It is based on the approximation of inverse of diagonal block matrices. In case $p = 1$, a similar form was already used in [2] for differential equations in two space dimensions. In [12] a special form for symmetric matrices was used.

Definition 2.1 Let H be a square matrix and let $p \geqslant 0$ be an integer. Then $[H]^{(p)}$ denotes the matrix with entries equal to those of H within the bandposition of H with halfbandwidth p and zero outside, i.e.,

$$[H]_{i,j}^{(p)} = \begin{cases} H_{i,j}, & |i-j| \leqslant p, \\ 0, & \text{otherwise.} \end{cases}$$

That $[H]^{(p)}$, $p \geqslant q$ is an accurate approximation of $H = G^{-1}$ if G has a small halfbandwidth q and a not large condition number H, follows by a theorem in [13]. In the symmetric positive definite case, it is shown that the entries of H^{-1} decay away from the main diagonal as $Cr^{|i-j|/q}$ for some positive constant C, where $r = (1 - \kappa^{-1/2})/(1 + \kappa^{-1/2})$ and $\kappa = \max \lambda(G)/\min \lambda(G)$. In Version 1 we let

$$\tilde{D}_0 = A_{1,1}, \qquad \tilde{D}_i = A_{i,i} - A_{i,i-1}\left[\tilde{D}_{i-1}^{-1}\right]^{(p)} A_{i-1,i}, \quad i = 2, 3, \ldots, n, \tag{2.9}$$

and in Version 2 we let

$$\tilde{D}_0 = 0, \qquad \tilde{D}_i = \left[\left(A_{i,i} - A_{i,i-1}\tilde{D}_{i-1}A_{i-1,i}\right)^{-1}\right]^{(p)}, \quad i = 1, 2, \ldots, n. \tag{2.10}$$

For the actual calculation of the entries of $[H]^{(p)} = [G^{-1}]^{(p)}$ we note that, if $p \geqslant q$,

(I) No entries outside the bandpart of $H = G^{-1}$ need to be calculated.

(II) If G is given on the form

$$G = (I - \tilde{L})B^{-1}(I - \tilde{U})$$

where B is block-diagonal and $\tilde{L}$, $\tilde{U}$ are strictly lower and upper block triangular, respective, then we do not need to calculate any inverses of block-matrices.

The first part was noted in [22] (see also [15]) and is based on the relations

(i) $G^{-1} = BL^{-1} + \tilde{U}G^{-1}$, where $L = I - \tilde{L}$,

(ii) $G^{-1} = U^{-1}B + G^{-1}\tilde{L}$, where $U = I - \tilde{U}$.

Relations (i) and (ii) are to be used for the calculation of the upper and lower triangular parts of G^{-1}, respectively. In this way L^{-1} and U^{-1} do not enter in the computation. For the extension of the method to the blockmatrix case the following algorithm was presented in [7]. We assume here for simplicity that $p = q$.

Algorithm ABI (approximate block inverse):

For $r = n, n-1, \ldots, 1$ do

$$(G^{-1})_{r,r} = B_{r,r} + \sum_{s=1}^{\min(p,n-r)} U_{r,r+s}(G^{-1})_{r+s,r}.$$

For $k = 1, 2, \ldots, p$ do

(i) $$(G^{-1})_{r-k,r} = \sum_{s=1}^{\min(p,n-r+k)} \tilde{U}_{r-k,r-k+s}(G^{-1})_{r-k+s,r};$$

(ii) $$(G^{-1})_{r,r-k} = \sum_{t=1}^{\min(p,n-r+k)} (G^{-1})_{r,r-k+t}\tilde{L}_{r-k+t,r-k}.$$

Note that only matrix-matrix multiplications occur. Further, if G is symmetric, then only one of (i) or (ii) is needed.

Positive definiteness

It has been proven in [2,8] and [10] that if the given matrix A is a nonsingular M-matrix, this property is shared by the partly factorized matrices that occur during the approximate factorization.

However note that even if a matrix B is positive definite, the approximation $[B]^{(p)}$ of B, as defined in Definition 2.1, need not be positive definite. This follows because $[B]^{(p)} = B - R$, where by definition, the defect matrix R is indefinite. Hence, if the smallest eigenvalue of the positive definite matrix B is small enough, the smallest eigenvalue of $[B]^{(p)}$ may be negative, causing $[B]^{(p)}$ to be indefinite.

It is interesting to note, however, that if we modify the diagonal part of $[B]^{(p)}$ so that the new approximation satisfies a (generalized) rowsum criterion (see Section 4) then R will be negative semidefinite in our applications, where the entries of B are non-negative. This means that in this case the approximation of B will preserve positive definiteness.

Anyhow, the possible lack of positive definiteness does not matter if the factorization is calculated as in Version 1, based on (2.9), because then the matrices $\tilde{D}_i$ will remain positive definite, even if $[\tilde{D}_{i-1}^{-1}]^{(p)}$ is indefinite as is easily seen. However, in Version 2, based on (2.10), we may loose positive definiteness (unless we modify the approximation as indicated above). Hence the corresponding preconditioning matrix C may be indefinite, which as well known will make the correspondingly preconditioned conjugate gradient method converge slower or even diverge.

Furthermore, in Section 5 it shall be indicated that Version 2 can be less robust then Version 1, in the respect that the corresponding preconditioner may be less accurate in many applications. On the other hand, we know that Version 2 is vectorizable, but Version 1 is not if we use forward and backward recursion.

To overcome this we propose to use the power expansion method as defined in (2.4) and (2.5) also for the linear block systems with matrix $\tilde{D}_i$. The resulting method then takes the following steps.

Version 3.

Step 1. Calculate an approximate factorization using algorithm (2.9).

Step 2. Factorize $\tilde{D}_i = (I - \tilde{l}_1)\tilde{d}^{-1}(I - \tilde{u}_1)$, $i = 2, 3, \ldots, n$ into lower and upper triangular factors, and where $\tilde{d}$ is diagonal.

Step 3. Expand the inverses of $\tilde{D}_i$, $i = 1, 2, \ldots, n$ into factors of powers,

$$\tilde{D}_i^{-1} = \tilde{\tilde{D}}_i = \prod_{k=0}^{s}\left(1 + \tilde{u}_1^{2^k}\right)\tilde{d}\prod_{k=0}^{s}\left(1 + \tilde{l}_1^{2^k}\right) \quad \text{or}$$

$$\tilde{\tilde{D}}_i = \left(I + \tilde{u}_1^{2^s}\right)\cdots\left(I + \tilde{u}_1\right)\tilde{d}\left(I + \tilde{l}_1^{2^s}\right)\cdots\left(I + \tilde{l}_1\right) \tag{2.11}$$

and keep the matrices on these forms. Alternatively, during the multiplication of $\tilde{\tilde{D}}_i$ with a vector, calculate the powers $l_1^{2^k} = (l_1^{2^{k-1}})^2$ as they are needed. Then we need storage only for $\tilde{u}_1$, $\tilde{d}$ and $\tilde{l}_1$.

Note that this method is most practical in case $p = 1$, when $\tilde{u}_1$ and $\tilde{l}_1$ has only one nonzero super- and sub-diagonal, respectively.

Step 4. Let $C = (I - L\tilde{\tilde{D}})\tilde{\tilde{D}}^{-1}(I - \tilde{\tilde{D}}U)$ where $\tilde{\tilde{D}} = \text{diag}(\tilde{\tilde{D}}_1, \tilde{\tilde{D}}_2, \ldots, \tilde{\tilde{D}}_n)$. Hence Version 3 is on division (inverse) free form, so no recursion appears within the blocks.

Remark 2.1. In many problems $\tilde{l}_1^{2^k}$ and/or $\tilde{u}_1^{2^k}$ are small compared to I if k is large enough. Hence we may discard some of the final factors in Step 3. This is most easily done when multiplying $\tilde{\tilde{D}}_i$ with the actual vectors, which occur during the iterations. Note then the order we take the factors in (2.10).

This method (Version 3) is vectorizable within the blocks. Corresponding to the power expansion method in (2.4) and (2.5) we may define a method, Version 4, where we use some approximation of the powers to keep loss of sparsity within reasonable limits. The corresponding method we call the approximate power expansion method, the APE-method. It will not further be discussed in this paper. This latter method is however fully vectorizable, both within the blocks as well as among the blocks.

In [21] another method is proposed which is vectorizable within each block. It is simply based on the approximation $\tilde{D}_i^{-1} \cong [\tilde{D}_i^{-1}]^{(p)}$ for each diagonal block matrix, during the forward and backward recursions. As is pointed out in [21], in this method one may loose positive definiteness. Again, we remark that we may modify the approximation to preserve positive definiteness.

3. Methods of nested approximate inverses and general block incomplete factorization for difference equations for three dimensional problems

Consider now the calculation of a sparse approximate factorization $C^{(3)}$ on the form of Version 2 in Section 2, where the matrix $A = A^{(3)}$ is of the form (2.6). Such a matrix arises in many discretized partial differential equation problems in three space dimensions, by a proper ordering of the unknowns. We shall show that we may readily apply the algorithm ABI for this purpose.

For the calculation of $C^{(3)}$ we need to calculate the sequence $\{\tilde{D}_i^{(2)}\}$ of block-diagonal matrices satisfying (2.8). This shall be done in two steps and by use of nested approximate inverses as follows:

For $r = 1, 2, \ldots, n$ do

Step 1: Calculate an approximate factorization G_r of $H_r = A_{r,r} - A_{r,r-1}\tilde{D}_{r-1}^{(2)}A_{r-1,r}$ of the form $G_r = (I - B\tilde{L})B^{-1}(I - B\tilde{U})$ where B is blockdiagonal, $B_{r,r} = \tilde{D}_r^{(1)}$ and where $\tilde{D}_0^{(2)} = 0$. Here $\tilde{D}_r^{(1)}$ is a sequence of line-block-matrices calculated as for a two space dimensional problem, i.e., for some $p_1 \geqslant 1$ we have $\tilde{D}_0^{(1)} = 0$, $\tilde{D}_i^{(1)} = [((H_r)_{i,i} - (H_r)_{i,i-1}\tilde{D}_{i-1}^{(1)}(H_r)_{i-1,i})^{-1}]^{(p_1)}$, $i = 1, 2, \ldots, m_r - 1$, where m_r is the order of H_r.

Step 2: Calculate the $p = p_2$ block bandwidth part of G_r^{-1} by use of the algorithm ABI of Section 2. Let

$$\tilde{D}_r^{(2)} = \left[\left[G_r^{-1}\right]\right]^{(p_2)}, \quad \text{where } \left[\left[G_r^{-1}\right]\right]_{i,j}^{(p_2)} = \begin{cases} (G_r^{-1})_{i,j}, & |i-j| \leqslant p_2, \\ 0, & \text{otherwise,} \end{cases}$$

and $(G_r^{-1})_{i,j}$ denotes the (i, j)th *block* of G_r^{-1}.

In this nested factorization way we calculate a sparse approximation $\tilde{D}_r^{(2)}$ of H_r^{-1}, with sparsity structure similar to that of a two space difference or finite element matrix. As was noted in Section 2, the effectiveness of this method will depend upon the condition number of the diagonal block matrices that occur in the matrix recursions. A study of these condition numbers has not yet been undertaken.

An alternative method for solving three-dimensional problems, which is based on the general incomplete block-matrix factorization method as presented in [4], will now be discussed. To this end, consider at first a difference matrix of the form which occurs in a *two*-dimensional problem, using the five-point difference method. Then the simplest version (IC(0)) of pointwise incomplete factorization takes the following form, when the parameter $\omega = 0$. For $\omega = 1$ we get the unperturbed modified incomplete factorization method (MIC(0), see [16]) which, for this problem, is identical to the generalized SSOR-method [1] in unperturbed form and to the DKR-method [14]. The influence of the relaxation parameter will be further discussed in Section 4.

We have in this case the set J of indices (i, j), where fill-ins will be permitted in the incomplete factorization, $J = \{(i, j),\ a_{ij} \neq 0\}$. Hence for a $m \times n$ mesh with rowwise orderings and say Neuman type boundary conditions (considering for instance the problem $-\Delta u + u = f$ on $\Omega = [0, 1]^2$, $\partial u/\partial n = 0$ or $\partial\Omega$) we have $J = \{(i, j),\ |i-j| = 1 \vee |i-j| = n \vee i,\ j = 1, 2, \ldots, N\}$ where $N = m \cdot n$.

Following the notations in [5] but presenting the algorithm in its relaxed form as in [9], we perform the following operations in an incomplete (and relaxed) factorization method:

Let $a_{ij}^{(1)} = a_{ij}$. For $r = 1, 2, \ldots, n-1$ do

$$l_{ir} = a_{ir}^{(r)}\left(a_{rr}^{(r)}\right)^{-1}, \tag{3.1a}$$

$$a_{ij}^{(r+1)} = \begin{cases} a_{ij}^{(r)} - l_{ir}a_{rj}^{(r)}, & (r+1 \leqslant j \leqslant N) \wedge ((i, j) \in J) \\ 0, & (r+1 \leqslant j \leqslant N) \wedge ((i, j) \notin J), \\ a_{ii}^{(r)} - l_{ir}a_{rj}^{(r)} + \omega \sum\limits_{\substack{p=r+1 \\ (i,p)\notin J}}^{N} \left(a_{ip} - l_{ir}a_{rp}^{(r)}\right), & j = i. \end{cases} \tag{3.1b}$$

where $i = r+1,\ r+2, \ldots, N$ and $0 \leqslant \omega \leqslant 1$. The matrices L and $A^{(r+1)}$, $r = 1, 2, \ldots, N-1$ are completely defined when we add

$$l_{ij} = \begin{cases} 0 & \text{for } j > i, \\ 1 & \text{for } j = i, \end{cases}$$

$$a_{ij}^{(r+1)} = \begin{cases} 0 & \text{for } j = 1, \ldots, r, \quad i = j+1, \ldots, N, \\ a_{ij}^{(i)} & \text{for } i = 1, \ldots, r, \quad j = i, \ldots, N. \end{cases}$$

The matrix U is defined by

$$U_{ij} = \begin{cases} 0 & \text{for } j < i, \\ a_{ij}^{(i)} & \text{for } i = 1, 2, \ldots, N, \quad j = i, i+1, \ldots, N. \end{cases}$$

The relaxed incomplete factorization RIC of A is then given by

$$C = LU. \tag{3.2}$$

Consider now the case where the entries a_{ij} are block-matrices such as would be the case for $A = A^{(3)}$, a 'three-dimensional' difference matrix with block property A. Then the corresponding RIC(0) incomplete factorization of $A^{(3)}$ would take the form (3.1), (3.2) with the exception that the inverses of the diagonal block matrices $a_{rr}^{(r)}$, which occur in (3.1a) will in general be full matrices and hence sparsity would be lost, which we will not permit. Hence, this is the point where we will apply the approximate block matrix inverse method of Section 2 (for line matrices, i.e., for $p = 1$). For the model type problems we are considering in this section, where the matrices $a_{i,i-1}$ *and* $a_{i,i+1}$ *are diagonal matrices, the line matrix* (*i.e., tridiagonal matrix*) *structure will then be preserved for* $a_{ii}^{(r+1)}$, $r = 1, 2, \ldots, N-1$. For a more general problem we may have to apply a further deleting of fillins also of each matrix block within the permitted block structure J. For a further discussion of this, see [4].

It follows from Theorem 3.1 in [4] that in case $\omega = 0$, the matrices $A^{(r)}$ remain M-matrices. In particular, $(a_{rr}^{(r)})^{-1} \geqslant 0$. Further, from Theorem 3.2 in that paper it follows that the splitting $A = C - R$, $C = LU$ as defined by (3.2) for $\omega = 0$, is a regular splitting. Hence the basic iterative method (2.1) is convergent.

Remark 3.1. The above method is presented on a form corresponding to Version 1 (diagonal block version). Similarly, we can define a general block matrix version, corresponding to Version 2 (diagonal block inverses version). In this case however one finds that the corresponding splitting is not a regular splitting. However, as we intend to use the matrix (as a preconditioner in some accelerated iterative method, this property is then not needed).

Remark 3.2. To show the improved performance a block incomplete factorization method may have, compared to classical methods, we shall at first compare it with the classical block iterative method of Jacobi type. To this end we consider the model test problem,

$$-\Delta_h^{(5)} u = f_h \quad \text{in } \Omega = [0, 1]^2, \qquad u = g_h \quad \text{on } \partial\Omega,$$

where $\Delta_h^{(5)}$ is the five-point difference operator.

We consider then the approximate factorization method of the type of Version 1 in Section 2, with the diagonal matrices defined by the recursion,

$$\begin{aligned} &\tilde{D}_i = A_{i,i} - A_{i,i-1}\left[\tilde{D}_{i-1}^{-1}\right]^{(p)} A_{i-1,i}, \quad i = 2, 3, \ldots, n, \\ &\tilde{D}_1 = \left[A_{1,1}^{-1}\right]^{(p)}, \qquad\qquad \text{with halfbandwidth } p = 1. \end{aligned} \tag{3.3}$$

Remark 3.3. With a sparsity structure J as chosen above the method will not change any of the offdiagonal blocks. For a problem of the form of a five-point difference matrix for the natural orderings, but where the matrix entries are themselves matrices; as is the case for a three-dimensional problem, the generalized incomplete block matrix factorization method is hence identical to the generalized SSOR method of Section 2. Then

$$C = (\tilde{D} - L)(I - \tilde{D}^{-1}U) = A + R$$

where

$$R = \tilde{D} + L\tilde{D}^{-1}U - D, \quad \text{where } D_i = A_{i,i}$$

and

$$R_{i,i} = A_{i,i-1}\left(\tilde{D}_{i-1}^{-1} - \left[\tilde{D}_{i-1}^{-1}\right]^{(p)}\right)A_{i-1,i} \geqslant 0, \quad \text{and } R_{i,j} = 0, \quad i \neq j.$$

As was already noted in this case, $C - R$ is a regular splitting of A. Hence the stationary-iterative method

$$Cx^{l+1} = Rx^l + b, \quad l = 0, 1, \ldots$$

converges and we have

$$x^{l+1} = Bx^l + C^{-1}b, \quad l = 0, 1, \ldots \tag{3.4}$$

where $B = C^{-1}R$. In order to get an idea of how accurate the preconditionings of the block incomplete factorization methods are, we now compare the spectral radius $\rho(C^{-1}R)$ with the corresponding one for the block Jacobi method.

In the latter method, the iteration matrix is $B = D^{-1}A$ and, for the above model problem we get $\rho(B) = \cos \pi h/(2 - \cos \pi h) \approx 1/(1 + (\pi h)^2)$, $h \to 0$ (see for instance [24]).

For $\rho(C^{-1}R)$ we get by a theorem in Varga [24] for regular splittings,

$$\rho(C^{-1}R) = \frac{\rho(A^{-1}R)}{1 + \rho(A^{-1}R)} \leqslant \frac{\|A^{-1}\|_2 \|R\|_2}{1 + \|A^{-1}\|_2 \|R\|_2}. \tag{3.5}$$

In order to estimate $\|R\|_2$ we consider the limit of the sequence (3.3) for the model problem. We get $\tilde{D}_i \to \tilde{D}$ where

$$\tilde{D} = D - [\tilde{D}^{-1}]^{(p)}, \quad D = \begin{bmatrix} 4 & -1 & & \\ -1 & 4 & -1 & \\ & \ddots & \ddots & \ddots \\ & & -1 & 4 \end{bmatrix}.$$

It is easy to see that the sequence $\tilde{D}_i^{-1}$ is increasing. We get $R_{i,i} = \tilde{D}_i^{-1} - [\tilde{D}_i^{-1}]^{(p)}$, $i = 1, 2, \ldots, n$. For $p = 1$ a calculation shows that $\|R\|_2 \cong 0.123$. Since $\|A^{-1}\|_2 \leqslant \frac{1}{2}(\pi h)^2$, we finally get from (3.5)

$$\rho(C^{-1}R) \leqslant 1/\big(1 + 1/\|A^{-1}\|_2 \|R\|_2\big) \approx 1/\big(1 + (4\pi h)^2\big), \; h \to 0.$$

Hence the number of iterations of the stationary iterative method (3.4) needed for a relative accuracy ε is about $(1/(\pi h)^2) \ln(1/\varepsilon)$, $h \to 0$ for the block Jacobi method and about $\frac{1}{16}(1/(\pi h)^2) \ln(1/\varepsilon)$ for the block incomplete factorization method (Version 1) for $p = 1$. The cost of the initial factorization of the latter method is about the same as the cost for one iteration step. Each iteration step costs less than twice as much as for the block Jacobi method. Hence the computational cost for the incomplete block factorization method is about $\frac{1}{8}$ of that for the block Jacobi method, i.e., very competitive.

Furthermore, both Versions 1 and 2 may be improved upon by 'modification' and this will be the topic of the next section.

4. Modifications of the incomplete block-matrix factorization method based on a generalized rowsum criterion

Definition 4.1. If $a_{i,j} \leqslant 0$, $i \neq j$ and if $Ac > 0$ for some positive vector c, $c > 0$, then A is called *generalized strictly diagonally dominant.*

It is known, see for instance [11], that a matrix A such that $a_{i,j} \leqslant 0$, $i \neq j$, is an M-matrix if and only if it is generalized strictly diagonally dominant.

In [4] it has been proved that for the general incomplete block-matrix factorization method, the modification of the incomplete factorization to preserve the generalized rowsums exists (the M-matrix property for the remaining-unfactorized part of the matrix is preserved at all stages of the factorization). Here we show that this result is also valid for the relaxed method if the modification is based on such a vector c, $c > 0$ for which $Ac > 0$.

For simplicity, we follow [9] and consider only the case of a tridiagonal block matrix. Then we have for Version 1, of the relaxed incomplete blockmatrix factorization,

$$\tilde{D}_1 = A_{1,1}, \qquad \tilde{D}_r = A_{r,r} - A_{r,r-1}\big[\tilde{D}_{r-1}^{-1}\big]^{(p)} A_{r-1,r} - \omega D_r', \quad r = 2, 3, \ldots, n, \tag{4.1}$$

where $0 \leqslant \omega \leqslant 1$ and D_r' is a diagonal matrix, so determined that

$$D_r' c = A_{r,r-1}\big(\tilde{D}_{r-1}^{-1} - \big[\tilde{D}_{r-1}^{-1}\big]^{(p)}\big) A_{r-1,r} c. \tag{4.2}$$

Note that the calculation of D_r' is inexpensive. In particular, the multiplication of the vector $(A_{r-1,r}c)$ by $\tilde{D}_r^{-1}$ is done by solving the corresponding linear system for the bandmatrix $\tilde{D}_r$.

We shall prove that the matrices $\tilde{D}_r$ in (4.1) which occur during the relaxed incomplete factorization method remain M-matrices and hence that the recursion exists.

Theorem 4.1. *Let A be a block tridiagonal M-matrix. Then the matrices $\tilde{D}_r$ defined by* (4.1) *are M-matrices and hence, in particular, nonsingular.*

Proof. By induction. Let $Ac = d$. Assume that $\tilde{D}_{r-1}$ is an M-matrix (which is the case for $r = 2$, because A and hence its diagonal blocks, are M-matrices). Consider the matrix

$$\begin{bmatrix} \tilde{D}_{r-1} & A_{r-1,\cdot} \\ A_{\cdot,r-1} & A^{(r)} \end{bmatrix},$$

where $A^{(r)}$ is the main submatrix part of A, that remains after elimination of the first $(r-1)$ blockrows and columns and where $A_{r-1,\cdot} = [A_{r-1,r}, 0, \ldots, 0]$ and $A_{\cdot,r-1}$ is similarly defined. For $r = 2$, This matrix is equal to A. Assume that

$$\begin{bmatrix} \tilde{D}_{r-1} & A_{r-1,\cdot} \\ A_{\cdot,r-1} & A^{(r)} \end{bmatrix} \begin{bmatrix} c_{r-1} \\ c^{(r)} \end{bmatrix} \geqslant \begin{bmatrix} d_{r-1} \\ d^{(r)} \end{bmatrix}, \tag{4.3}$$

where c_{r-1}, d_{r-1} are the $(r-1)$st block components and $c^{(r)}$, $d^{(r)}$ consists of the last $n - r + 1$ blockcomponents of c and d, respectively. Because $Ac = d$, (4.3) is valid for $r = 2$. By elimination of c_{r-1} in (4.3) we get

$$\begin{bmatrix} B_{r,r} & A_{r,\cdot} \\ A_{\cdot,r} & A^{(r+1)} \end{bmatrix} \begin{bmatrix} c_r \\ c^{(r+1)} \end{bmatrix} \geqslant \begin{bmatrix} d_r - A_{\cdot,r-1}\tilde{D}_{r-1}^{-1} d_{r-1} \\ d^{(r+1)} \end{bmatrix} \geqslant \begin{bmatrix} d_r \\ d^{(r+1)} \end{bmatrix} \tag{4.4}$$

where $B_{r,r} = A_{r,r} - A_{r,r-1}\tilde{D}_{r-1}^{-1} A_{r-1,r}$. By (4.1) and (4.2), we have $\tilde{D}_r c_r = B_{r,r} c_r + (1 - \omega) D_r' c_r$ Since by definition, $0 \leqslant [\tilde{D}_{r-1}^{-1}]^{(p)} \leqslant \tilde{D}_{r-1}^{-1}$, we have by (4.2) that $D_r' c \geqslant 0$. Hence, since $\omega \leqslant 1$, $\tilde{D}_r c_r \geqslant B_{r,r} c_r$ so by (4.4)

$$\begin{bmatrix} \tilde{D}_r & A_{r,\cdot} \\ A_{\cdot,r} & A^{(r+1)} \end{bmatrix} \begin{bmatrix} c_r \\ c^{(r+1)} \end{bmatrix} \geqslant \begin{bmatrix} d_r \\ d^{(r+1)} \end{bmatrix} > 0.$$

Since the offdiagonal entries of $\tilde{D}_r$ are nonpositive, it follows that

$$\begin{bmatrix} \tilde{D}_r & A_{r,\cdot} \\ A_{\cdot,r} & A^{(r+1)} \end{bmatrix}$$

is generalized strictly diagonally dominant and hence an M-matrix. In particular, $\tilde{D}_r$ is a nonsingular M-matrix and by induction, the theorem is proven. □

Remark 4.1. The relaxation with $\omega = 1/(1 + \zeta_1 h^2)$ or $\omega = 1/(1 + \zeta_2 h)$, $\zeta_1, \zeta_2 > 0$ has the same effect as the perturbations applied in [1], [16] and [5]. These perturbations where applied in order to prove that the spectral condition number after preconditioning of a second order difference matrix is decreased by an order of magnitude.

The vector c in (4.2) may be chosen in many ways. For instance, if (an approximation of) the first eigenvalue $v^{(1)}$ of A is known, then $c = e + \zeta v^{(1)}$, for some $\zeta > 0$, where $e = (1, 1, \ldots, 1)^T$ is a good choice, if $Av^{(1)} > 0$. The relaxation parameter ω has an interesting effect on the distribution and clustering of the spectrum of $C^{-1}A$ as we shall see in the next section.

5. Eigenvalue distribution and superlinear rates of convergence of the conjugate gradient method

Consider an elliptic problem discretized to get a linear system with an M-matrix. It is easily understood that for the relaxed method with $\omega = 0$ (i.e., the classical incomplete factorization method such as the method in [19], the smooth eigenvectors (first harmonics) of $C^{-1}A$ correspond to small eigenvalues and the rough or 'noisy' ones correspond to eigenvalues close to 1. Since the number of smooth eigenvectors is relatively small, this indicates that the spectrum is sparse for small eigenvalues and this is indeed found in practice, see [20] and [12].

For the classical modified incomplete factorization method in [16], one finds on the other hand that for smooth eigenvectors v, $Cv \cong Av$, i.e., the corresponding eigenvalues of $C^{-1}A$ are close to 1. For the same reason as for the unmodified method, the 'noisy' eigenvectors correspond to eigenvalues close to 1. It is only the 'average' fast modes (with frequency $\approx h^{-1/2}$) which give eigenvectors far away from 1. It is further known (see for instance [5]) that all eigenvalues of $C^{-1}A$ are larger or equal to 1. For the relaxed method with $0 < \omega < 1$, one can expect to find both small and large discrete eigenvalues and clustering about 1. Further one finds that for the incomplete factorization methods for block matrices, the spectrum is both more clustered about 1 and has much smaller condition number than for the pointwise methods for this test problem.

Further results and discussions about this are found in [9]. In that paper it is also shown that such eigenvalue distributions are very favorable for the rate of convergence of the conjugate gradient method. In particular, it is shown, see also [17], that for the case of an iteration matrix $C^{-1}A = I + B$, where B is compact (i.e., has eigenvalues with cluster point at 0) the rate of convergence of the conjugate gradient method is superlinear, i.e., the iteration errors satisfy $\|e^{(k)}\| \leqslant (\rho_k)^k \|e^{(0)}\|$, $k = 1, 2, \ldots$ where the average convergence factor $\rho_k \to 0$ as $k \to \infty$. With $k \to \infty$ we mean that k gets large, but still k is much smaller than the order of the matrix. Typically $k = O(h^{-1/2})$ for a difference equation. Note that for a uniform distribution of eigenvalues we have $\rho_k \approx (1 - \kappa^{-1/2})/(1 + \kappa^{-1/2}) > 0$, $k \to \infty$, where $\kappa = \max \lambda(C^{-1}A)/\min \lambda(C^{-1}A)$ where $\lambda(\ )$ indicates the eigenvalue function.

This means that the conjugate gradient method for preconditionings with a spectrum such as for the relaxed methods eventually will converge faster than even iterative methods of optimal order, like the multilevel method presented in [27]. That method has a computational complexity of $O(N \log N)$.

The actual condition numbers as found in [9] and [27], can be compared in Table 5.1. The asymptotic behavior of the condition numbers when $h \to 0$ has not yet been derived.

Note that the very impressive reductions of the condition numbers for the RIC and RBIC methods for $\omega = 1$, occur already for such a coarse mesh as 32×32.

6. Robustness of the incomplete block-matrix factorization methods

The numerical results reported in the previous sections were for the model problems. It is interesting to discuss what will happen for more general problems. In [9] a problem with discontinuous coefficients is treated and it is found that the incomplete block matrix factorization (Version 1) performs essentially as well in this case as for the model test problem.

We shall now heuristically discuss what can be expected for problems with unisotropy (6.1) and with convection dominated convection-diffusion (6.2), discretized by central and upwind differences, respectively.

$$\varepsilon u_{xx} + u_{yy} = f, \quad 0 < \varepsilon \ll 1, \tag{6.1}$$

$$-\varepsilon \Delta u + u_y = f, \quad 0 < \varepsilon \ll 1. \tag{6.2}$$

For the RBIC (block) methods, we let the diagonal blocks correspond to differences in the x-direction (i.e., we choose an ordering along the x-direction). Then the diagonal blocks will be strongly diagonally dominant and hence the ABI-approximation of the inverses will be accurate. This applies to both Versions 1 and 2.

At each stage of the matrix recursion we calculate

$$\tilde{D}_i = A_{i,i} - A_{i,i-1}\left[\tilde{D}_{i-1}^{-1}\right]^{(p)} A_{i-1,i} \qquad \text{(Version 1)} \tag{6.3}$$

$$\tilde{D}_i = \left[\left(A_{i,i} - A_{i,i-1}\tilde{D}_{i-1}A_{i-1,i}\right)^{-1}\right]^{(p)} \qquad \text{(Version 2)}. \tag{6.4}$$

Now if the diagonal blocks $A_{i,i}$ have large condition numbers typically which is the case if they

Table 5.1
Spectral condition number for A_h, the RIC (relaxed pointwise incomplete factorization) method, the RBIC (relaxed blockwise incomplete factorization) method (Version 1) and the hierarchical basis function method (h-version) for the model test problem.

h^{-1}	A_h	RIC $\omega = 0$	RIC $\omega = 1$	RBIC $\omega = 0$	RBIC $\omega = 1$	hierarchical b.f. h-version
16	103	10	4.48	2.5	1.60	19.53
32	414	40	9.33	7.8	2.78	31.85
64	1659		19.51		5.29	47.14
128	6640					65.38
∞	$\sim (4/\pi^2)h^{-2}$	$\sim \frac{4}{100}h^{-2}$	$\sim 0.3h^{-1}$		$\sim 0.08\, h^{-1}$	$O((\log h^{-1})^2)$

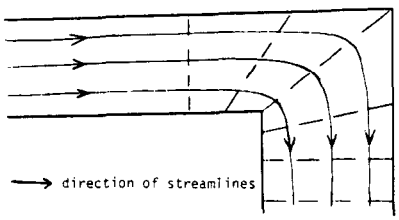

Fig. 1.

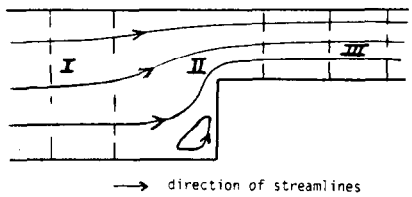

Fig. 2.

are not 'strongly' diagonally dominant, then $[\tilde{D}_{i-1}^{-1}]^{(p)}$ is a bad approximation. However, because in problems like (6.1) and (6.2) (with a columnwise ordering) for instance, then $\|A_{i,i-1}\| = O(\varepsilon)$ and $\|A_{i-1,i}\| = O(\varepsilon)$, so the term $A_{i,i-1}\tilde{D}_{i-1}^{-1}A_{i-1,i}$ has a norm of $O(\varepsilon)$. Hence $\tilde{D}_i$ in (6.3) is still an accurate approximation of $A_{i,i}$ and of the exact Schur complement, $A_{i,i} - A_{i,i-1}\tilde{D}_{i-1}^{-1}A_{i-1,i}$. This applies to Version 1.

For Version 2, on the other hand, the approximation $\tilde{D}_1 = [A_{1,1}^{-1}]^{(p)}$, etc. will actually also be used in the forward and backward solution processes. This implies that the incomplete block matrix factorization, Version 1 is more robust than Version 2 and than the RIC (pointwise) method.

For a vector computer, because Version 2 is better vectorizable than Version 1, one may recommend a combination of both versions, where Version 2 will be applied in regions where it is possible to apply a favorable direction of orderings of the unknowns. Alternatively, we may apply Version 3.

We give below two examples where it is possible to choose such a favorable direction of orderings.

Example 6.1. Convection-diffusion in an L-shaped domain, a 'knee' (see Fig. 1). Here we order the nodepoints along the dashed lines (which are approximately orthogonal to the streamlines). This will work like a marching along the directions of the characteristic lines for the corresponding Euler equation.

Example 6.2. Convection-diffusion in a 'duct' (see Fig. 2). In sub-regions I and III we use a columnwise ordering of the nodepoints (along the dashed lines). In region II we recommend to use Version 1.

References

[1] O. Axelsson, A generalized SSOR method, *BIT* **12** (1972) 443–467.

[2] O. Axelsson, S. Brinkkemper and V.P. Il'in, On some versions of incomplete block-matrix factorization iterative methods, Report 8322, Department of mathematics, Catholic University, Nijmegen, The Netherlands, May 1983.

[3] O. Axelsson, S. Brinkkemper, and V.P. Il'in, On some versions of incomplete block-matrix factorization iterative methods, *Linear Algebra Appl.* **58** (1984) 3–15.

[4] O. Axelsson, A general incomplete block-matrix factorization method, Report 8337, Department of Mathematics, Catholic University, Nijmegen, The Netherlands, October 1983, *Linear Algebra Appl.*, to appear.

[5] O. Axelsson and V.A. Barker, *Finite Element Solutions of Boundary Value Problems, Theory and Computation* (Academic Press, New York, 1984).

[6] O. Axelsson, A survey of vectorizable preconditioning methods for large scale finite element matrix problems CNA-190, February 1984, Center for Numerical Analysis, University of Texas, Austin, TX.
[7] O. Axelsson, A vectorizable variant of a block incomplete factorization method, *Proceedings of the VIth International Conference on the Numerical Solution of Fluid Flow Problems, Jan. 1984, University of Texas* (Wiley, New York) to appear.
[8] O. Axelsson, On incomplete factorization methods for block-matrices, presented at the International Conference on Numerical Analysis, March 19–21, 1984, Munich.
[9] O. Axelsson and G. Lindskog, On the eigenvalue distribution of a class of preconditioning methods, in preparation.
[10] R. Beauwens and M. Ben Bouzid, On block-OBV methods, Part 1 and 2, Scientific Report, CEN, Algiers (1983).
[11] A. Berman and R.J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences* (Academic Press, New York, 1979).
[12] P. Concus, G.H. Golub, and G. Meurant, Block preconditioning for the conjugate gradient method. Report LBL-14856, Lawrence Berkeley Laboratory, University of California, 1982; *SIAM J. Sci. Statist. Comput.*, to appear (abridged).
[13] S. Demko, W.F. Moss, and P.W. Smith, Decay rates for inverses of band matrices, *Math. Comput.*, to appear.
[14] T. Dupont, R.P. Kendall and H.H. Rachford, A approximate factorization procedure for solving self-adjoint elliptic difference equations, *SIAM J. Numer. Anal.* **5** (1968) 559–573.
[15] A.M. Erisman, and W.F. Tinney, On computing certain elements of the inverse of a sparse matrix, *Comm. ACM* **18** (1975) 177–179.
[16] I.A. Gustafsson, A class of first order factorization methods, *BIT* **18** (1978) 142–156.
[17] R.M. Hayes, Iterative methods of solving linear problems on Hilbert space, *Appl. Math. Sci.* **39** (1954) 71–103.
[18] D.R. Kincaid, T.C. Oppe, and D.M. Young, Vector computations for sparse linear systems, Report CNA-189, Center for Numerical Analysis, University of Texas, Austin, TX, February 1984.
[19] J.A. Meijerink and H.A. Van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Math. Comput* **31** (1977) 148–162.
[20] J.A. Meijerink and H.A. Van der Vorst, Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems, *J. Comput. Phys.* **44** (1981) 134–155.
[21] G. Meurant, The block preconditioned conjugate gradient method on vector computers, *BIT*, to appear.
[22] K. Takahishi, J. Fagan, and M.S. Chen, Formation of a sparse bus impedance matrix and its application to short circuit study, Proc. 8th PICA COnference 1973, Minneapolis, MN, 63–69.
[23] R.S. Varga, Factorizations and normalized iterative methods, in: R.W. Langer, Ed., *Boundary Problems in Differential Equations* (University of Wisconsin Press, Madison, WI, 1960) 121–142.
[24] R.S. Varga, *Matrix Iterative Analysis* (Prentice Hall, Englewood Cliffs, NJ, 1962).
[25] Z. Woźnicki, Two-sweep iterative methods for solving large linear systems and their application to the numerical solution of multigroup multi-dimensional neutron diffusion equation, Doctoral Dissertation, Report no. 1447/CYFRONET/PM/A, Institute of Nuclear Research, Swierk, Poland, 1973.
[26] D.M. Young, *Iterative Solution of Large Linear Systems* (Academic Press, New York, 1971).
[27] H. Yserentant, On the multilevel splitting of finite element spaces, Bericht Nr. 21, 1983, Institut für Geometrie und Praktische Mathematik, R.W.T.H., Aachen, Germany.